https://ntrs.nasa.gov/search.jsp?R=20190018055 2019-08-31T13:35:01+00:00Z

Designing and Building a Crew-Centric Mobile Scheduling and Planning Tool for Exploring Crew Autonomy Concepts Onboard the International Space Station

S. R. Hillenius, J. Marquez, I. Deliz, B. Kanefsky, D. Korth, M. Healy, S. Gibson, J. Zheng

Steve Hillenius
Playbook Lead
steven.r.hillenius@nasa.gov
NASA Ames Research Center

Technical Briefing for the ISS R&D Conference 2016

Motivation

To design and develop a technology platform to enable assessment of critical questions about the feasibility and effectiveness of crew self-scheduling and onboard planning:

Can onboard planning be performed autonomously and safely, while not significantly impacting crew workload and operational throughput?

Can we build a technology platform to empower crew and at the same time be easy to use?

Can we demonstrate that this technology platform works on ISS with real plans and procedures?

What is Playbook?

Playbook is a next generation easy-to-use mobile web-based planning tool.

Features include collaborative self-scheduling with constraint checking and violation visualizations, full activity execution status capabilities, built in communications chat functionality (text, photo, video, or files), condition band support, task list support, procedure linking support, full support for time-delay simulations and synchronization across multiple servers.

No installation is required, simply navigate to the playbook link inside your web browser.

Playbook Feature Overview

Current Features Include:

- **Collaborative self-scheduling functionality with ScratchPad (staging) support***
- **Two-server ISS synchronization solution (works during LOS)***
- Web-based plan viewer timeline, designed for tablet use
- **Can directly link and open to native IPV XML Procedures***
- PDF procedure viewer
- Mission log communications with photo and video support
- Activity execution status and activity note support
- **Activity flight rule constraint visualizations***
- **Fully simulated time-delay support***
- **Violation detection support***
- **Condition bands (S, Ku, day/night profiles, etc)***
- **Cyrillic Support***
- **Reads in and displays full ISS plans from Score***
- **Full time delay simulation support***
- **Unobtrusive data collection support***
- HTTPS Encrypted Transit and Connections
- **Task list support***
- Groups support

**Indicates new features built specifically to support the ISS deployment*

Thoughts for use of Playbook

- Use Playbook to:
  - Visualize mission timelines
  - Status Activity execution (active, completed, deferred, etc.) from schedule or task list
  - schedule activities from the task list
  - reschedule flexible activities
  - Access procedures/supporting documents for an activity
  - Write activity notes

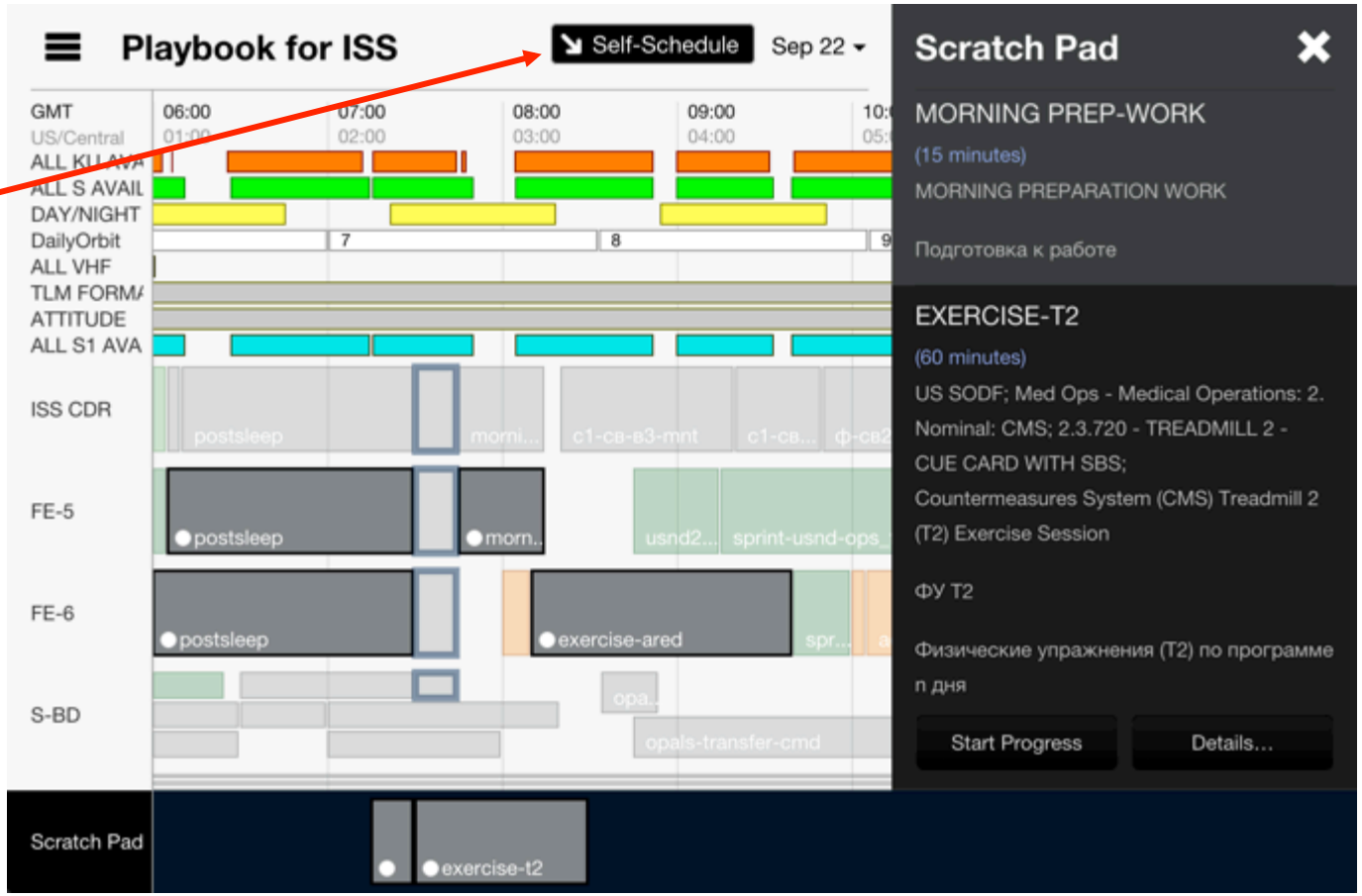
- Explore use of the mission log (built in chat and activity note functionality) to convey additional information (pictures, additional text, etc.) to the ground team

- Perform Lightweight Plan Editing (rescheduling activities, trying out what-if scenarios etc.)

Playbook Feature Overview: Self-scheduling Support

To self-schedule an activity, simply click on the self-schedule button on the timeline

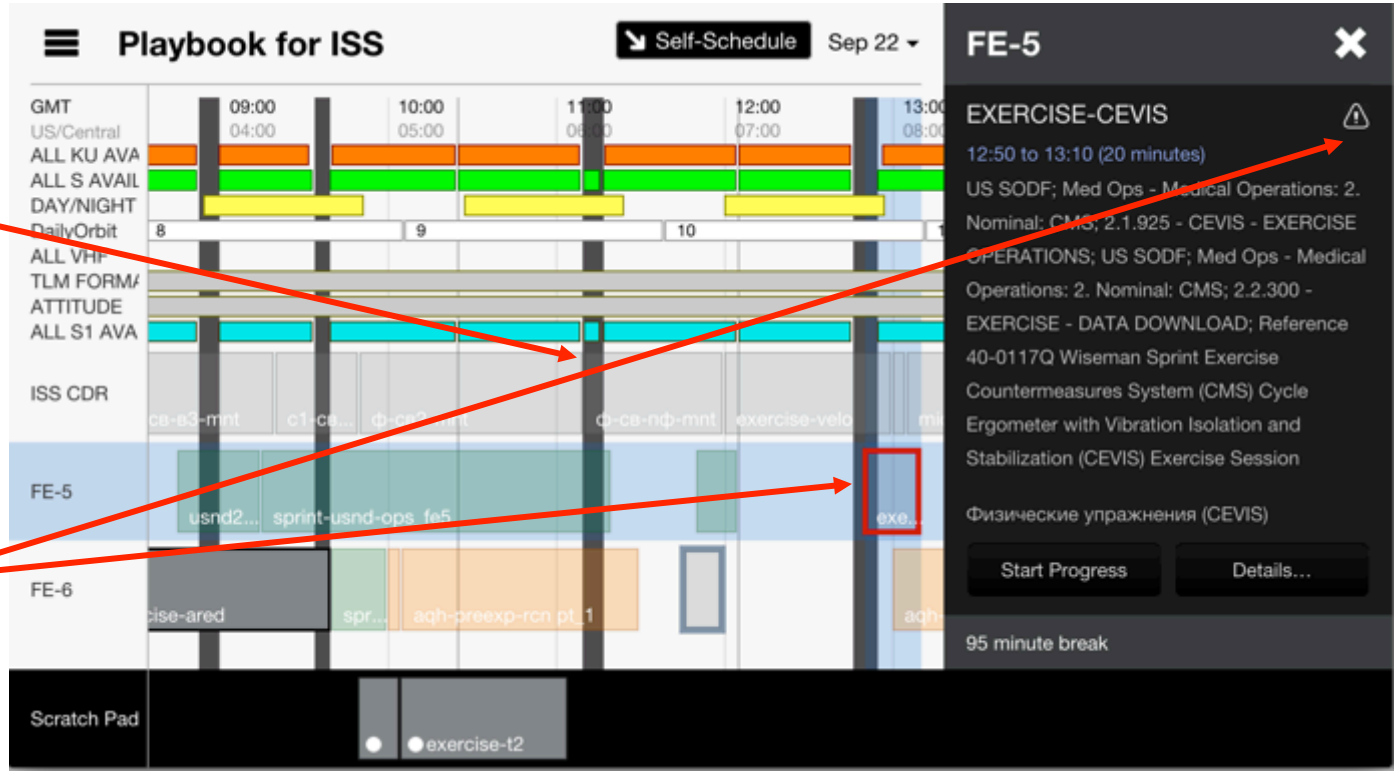
Rescheduling an activity is as simple as dragging and dropping the activity on the timeline



Playbook Feature Overview: Constraint Visualization and Violation Detection

Playbook visualizations temporal, resource and activity constraints as go/no-go zones

If an activity is moved into a no-go zone, Playbook outlines the activity in red and shows it in violation



Development Timeframe for ISS Adaptation and Deployment

About 1yr to adapt, test, certify and deploy on board ISS

Significant software development required to adapt Playbook (~6 months)

Formalized testing (3 months)

Service Pack testing (3 months)

Uplink to ISS (Set by the program)

Ground Checkout (Set by the program)

Crew Checkout (Set by the program)

Technical Challenges

Collaborative self-scheduling on a mobile device

Synchronization between tablet web clients on ISS and on the ground

Ensuring performance on onboard ISS iPads, tablets, and SSC Laptops

Adapting our software to work on board ISS and support ISS operational products and processes

End to end testing, certification

Web app, native app, or hybrid app?

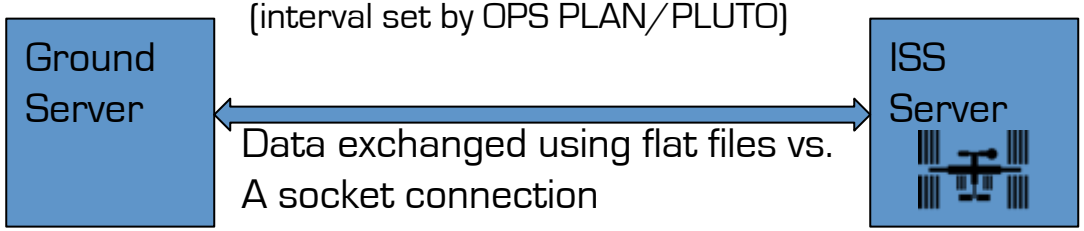
Playbook Feature Overview: Two-server Synchronization ISS Adaptation

Playbook is designed to be fully used during LOS.

Once in AOS again, all Playbook data is automatically synchronized between the servers.

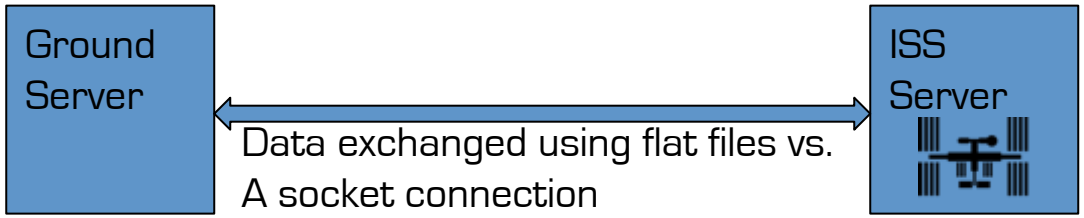
This functionality can also be used to automatically synchronize to a backup or mirrored server.

During AOS, all playbook data is synced in near real time
(interval set by OPS PLAN/PLUTO)



During LOS, crew can fully use playbook without a connection to ground

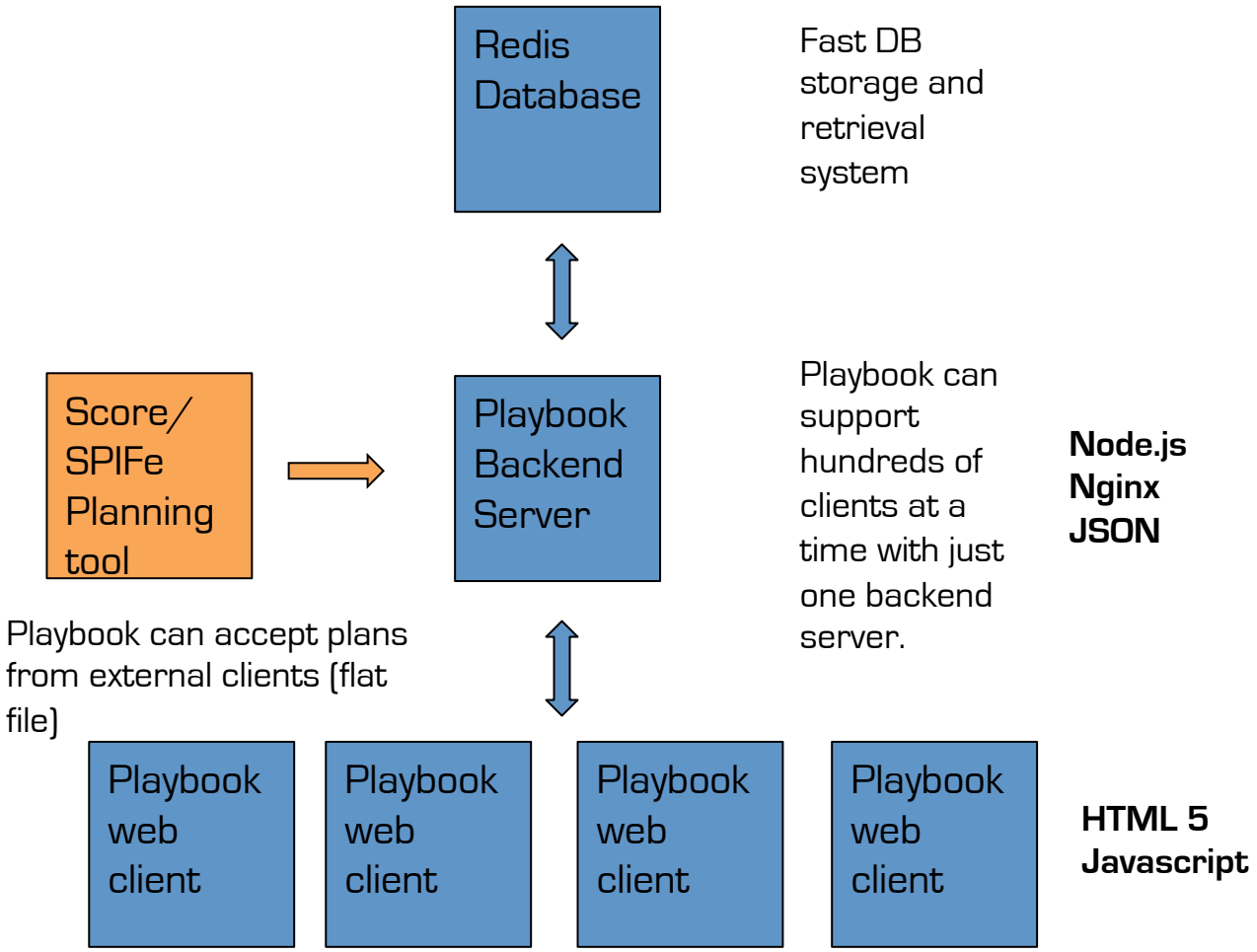
Once in AOS again, the connection is reestablished and all Playbook data is automatically synchronized between the servers.



Playbook Feature Overview: Software Architecture

Playbook is a modern web based application, designed for both touch and desktop devices.

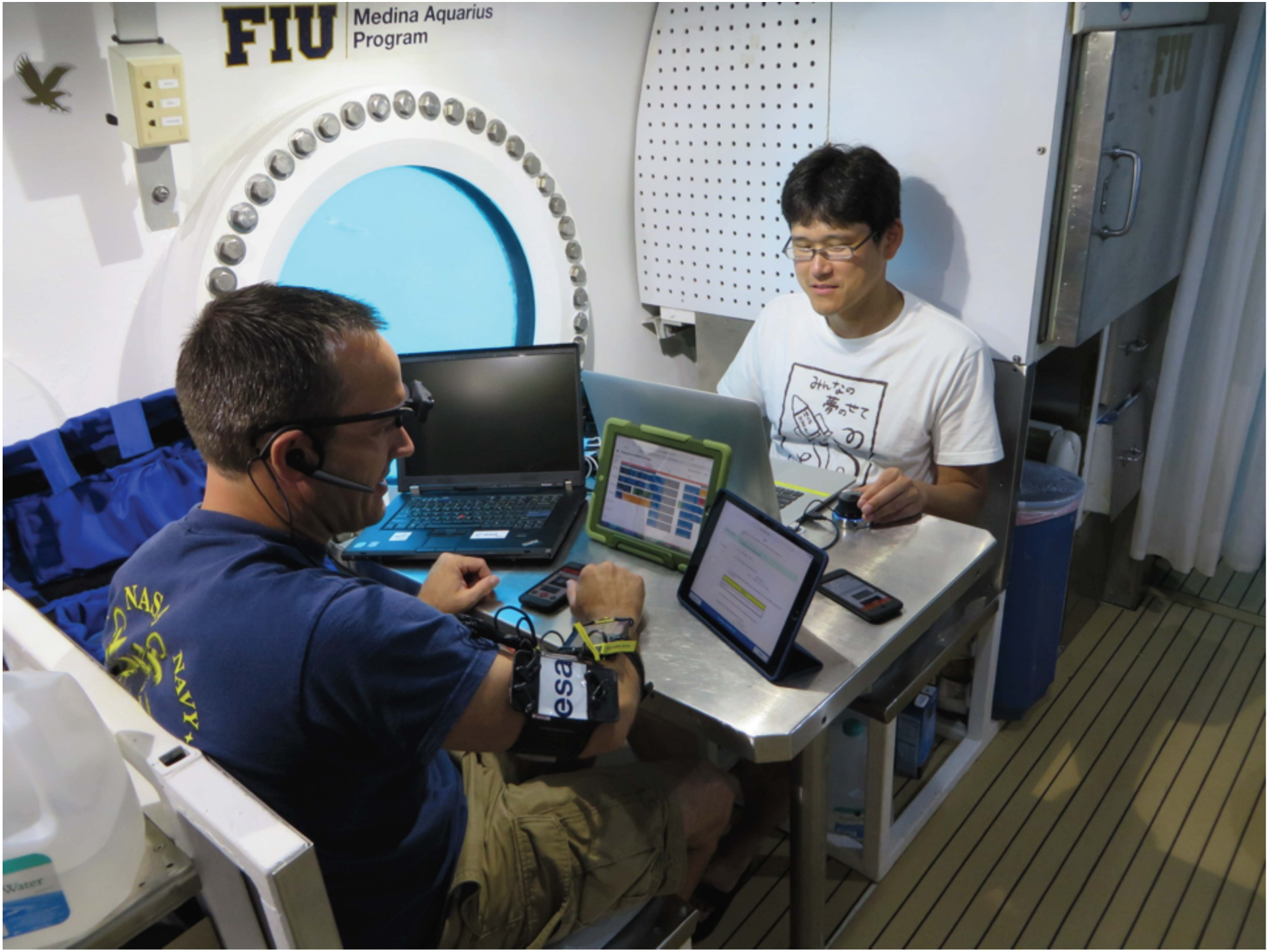
Playbook uses technologies such as HTML5 Javascript, node.js, NGINX,and Redis to allow fast interactive performance in the web browser.



Playbook front-end code is written for modern web-browsers such as Firefox, Chrome, and Safari. No installation of software is required, users simply type in the Playbook link into their web browser. Clients are continuously synchronized as long as they have a connection to the server. If they lose connection, they are synchronized once reconnected.

Playbook Live Demo


FIU
FIU


FIU
Medina Aquarius
Program
FIU

Questions?

Thanks!

Acknowledgments

This work was funded by HRP and the ISS Program.